

REMARKS/ARGUMENTS

In the Office Action dated April 7, 2004, the Examiner rejected all pending claims over the teachings of US Patent 6,473,767 granted to Bailey either alone or in combination with US Patent 5,764,972 granted to Crouse.

In the above-identified Office Action, Applicant submits that the Examiner has made omnibus rejections of numerous dependent claims in a summary fashion without taking into account the specific limitations recited in each individual dependent claim. Applicant is filing this Request for Continued Examination to ensure thorough examination of all claims. Applicant notes that dependent claims don't necessarily stand and fall with their respective independent claims. **The Examiner is requested to consider each dependent claim on its own merit, including all its limitations as required by 35 USC 112, 4th paragraph.**

Moreover, as discussed towards the end of this Amendment, Examiner's rejection of the independent claims is also meritless, in view of the lack of prior art.

Claim 14 is now written in independent form. Support for Claim 14 in its current form can be found in originally-filed Claims 1, 14 and 18. Claim 14 now requires a copy operation to send an email message to a user that starts the method, if a resource at a destination is full. In the above-identified Office Action, the Examiner rejected Claims 14 and 18 with an explanation in the third paragraph on page 5 of the above-identified Office Action. Applicant submits that the Examiner's rejection is completely without basis in the prior art cited by the Examiner. Specifically, to reject Claim 14, the Examiner cited only column 7, lines 31-67 and column 8 lines 1-65 of Bailey's patent. The cited text is reproduced below:

Column 7 lines 31-67:

Each file is then copied from source to target in the manner illustrated in FIG. 3.

If source file 300 is an anti-file, such as B*, as determined from its N/A attribute field and a target file entry exists (301), it is copied over the target file 302, such as B. If file 302 is a real file, the source anti-file type is checked (303) from the anti-file count field, m, to determine if it is transient (m=1) or persistent (m>1). If transient, file 302 (B in the above example) is deleted (304), and the anti-file B* is not written to the target directory.

If the anti-file is persistent, the target file B is still deleted (304) but the anti-file B* is written to in the target file directory, marked as an anti-file (305). To complete the description of the right hand side of FIG. 3, the two other cases which might arise are that there was no target file of the same name, in which case a target anti-file entry is simply created as a result of the copy operation (307) or the target file was already an anti-file, in which case the new anti-file entry is copied over it. In these two cases, the resulting target file directory entry is marked as an anti-file (306) by setting the appropriate attribute to A.

Turning now to the left hand side of FIG. 3, if the source file was a real file to be copied, such as A' or C in the above example, then its data would be read (310). If no target file exists (311), a directory entry is created (316). If a target file does not exist and is real (312) then the data read at 310 is written to data storage (317). If the target file entry was indicated to be an anti-file at 312, the data read at 310 would be discarded (313) and the anti-file type determined (314). If transient, the anti-file would be deleted (315) or if persistent, would remain in the target directory. If the target file was real, the new data pointed to by A' or C would be written to disk (315).

The invention as described above may be applied to a build library process for program source code development. The development of long and complex computer programs

Column 8 lines 1-65:

or other machine readable material involves the creation of source material, often in separate modules, and its repeated editing and revision through multiple iterations. Keeping track of these iterations so that the eventual document represents the most current version is a problem, particularly if there are multiple authors of the separate parts. Eventually, the final version must be built from the correct and current parts.

A common way of storing source code for a product is in a library system which supports at least the operations of storing new or changed source parts into the system, retrieving a specific level of source part, or querying and reporting on the parts in the system.

The changes to source code during the development process are often maintained by the library system in such a way that a particular level or set of changes can be extracted from the library as input to the process used to build the product.

The build process consists of the operations of compiling, linking and otherwise processing the input source to create the executable code in the shipped product. The build process is not necessarily carried out on the same operating system or physical machine as the library system, in some cases the product source for more than one eventual target operating system is extracted from the library system and built differently on each target system.

In order to build the relevant source parts for a product, it is necessary to get the correct level of those source parts in a location where the build processes can use them. This shadowing process is where problems start to arise. For performance reasons it is attractive to move only the new or changed data to the build shadow, yet the contents of the build shadow must reflect exactly the set of source which is intended to be built. If a source part is no longer present in the library then the shadowed counterpart of that source part must be deleted.

The above described implementation of the invention may be used to handle these repeated copy and delete operations in such a way that the build shadow is always current and contains no files which have been deleted from the source.

Another application of the invention as implemented herein is to an install process for a software package. The main aim of an install process for the executable and associated files of a software package is to provide the complete set of files necessary to run and administer that package.

If the package is already installed on the target system when this install process runs it must make sure that none of the files from the older version of the package are left in situ, since these would cause confusion or even errors. For example, old files containing help information may be referred to by the user because they are visible in the system, thus causing confusion.

Install processes commonly use the locally-held information about the prior version to use the 'uninstall' process of that system to clear up the old version before installing the new material. This depends on the system having information about the package version so that the appropriate files can be deleted. If this information is damaged or incomplete due to ill-advised attempts to remove the package manually then the install process will not be able to delete the appropriate files before the new install. The information might be unusable because it is for an unsupported version of the package (e.g. a beta-test version which did not install using the conventional mechanism).

As seen by reading the above-quoted text from Bailey's patent, there is no support whatsoever for email, contrary to the Office Action.

In fact the word "email" doesn't appear anywhere in the entirety of Bailey's patent. For this reason, Applicant repeats a request made in the prior amendment dated October 6, 2003, at the bottom of page 12. Specifically, the Examiner is requested to provide a reference which discloses or suggests sending of an email message if a resource at a destination is full.

There is also no support in the above-quoted Bailey's text for the Examiner's statement at the end of the third paragraph on page 5 of the Office Action, "Note: email messages system are used for file transferring processing." Firstly, Applicant notes, for the record, that the Examiner's Note is not in grammatically correct English language. Secondly, assuming that the "Note" is saying that there are prior art systems in which email messages are used to perform file transfer, Applicant requests the Examiner to supply such prior art references, as per MPEP 2144.03. Thirdly, even if the Examiner's Note is supported by a prior art reference, Applicant submits that the Examiner's Note is insufficient to support a *prima facie* rejection of Claim 14 at least because Claim 14 does not require that email be used for file transfer. Instead, Claim 14 requires that if, during copying, a resource at a destination is found to be full, then an email message is sent.

To the extent that the Examiner believes that it is inherent to send an email whenever the destination becomes full, Applicant respectfully requests the Examiner to identify, in the next Office Action, what text in Bailey's patent necessarily leads to such a conclusion. The Examiner is requested to provide a pin point cite to a specific line or a handful of lines of Bailey's patent or in any other prior art reference.

In the rejection of Claim 14, the Examiner did not provide any motivation whatsoever for the modification proposed in his "Note". For this reason, the Examiner is hereby requested to explain why a skilled programmer would be motivated to send an error message by email instead of simply displaying the error message on a screen. In view of the above remarks, Applicant submits that Claim 14 is in form for allowance, and allowance thereof is respectfully requested.

Claim 15 was rejected over the above-quoted text from Bailey's patent. As seen from the text quoted at page 7 of this amendment, there is no indication whatsoever about a copy operation waiting to be restarted. Therefore, even if email messages of the type recited in Claim 14 are described in Bailey's patent (although as noted above, they are not), Claim 15 provides an independent ground for patentability. If Claim 15 is rejected, then per MPEP 2144.03, please cite a reference for restarting of copy operation.

Claim 16 was also rejected over the above-quoted text from Bailey's patent. As seen from the text quoted at page 7 of this amendment, since there no indication of waiting to be restarted, the issue of how such waiting is to be implemented doesn't arise. Bailey's

patent fails to disclose or suggest sending a signal to self to suspend execution.

Therefore, the Examiner has failed to make a prima facie case of rejection of Claim 16.

Per MPEP 2144.03, please cite a reference for sending a signal to self.

Claim 17 was also rejected over the above-quoted text from Bailey's patent. As seen from the text quoted at page 7 of this amendment, since there is no indication of waiting to be restarted, hence the issue of what happens on being restarted doesn't arise. Bailey's patent fails to disclose or suggest recopying from the beginning. Therefore, the Examiner has failed to make a prima facie case of rejection of Claim 17. The Examiner is requested to explain why any prior art copy operation does not simply resume where it left off, instead of recopying as recited in Claim 17. If Claim 17 is rejected, then per MPEP 2144.03, please cite a reference for recopying from the beginning on being restarted.

Claim 18 was rejected over the above-quoted text from Bailey's patent. As seen from reviewing the above-quoted text, there is no indication whatsoever about sending email messages. Therefore, there is no issue of how to find an email address to which the email message is to be sent. Bailey's patent fails to disclose or suggest how to find email addresses. There is certainly no indication whatsoever of looking up a password file in Bailey's patent. Once again, the Examiner has failed to make a prima facie case of rejection of Claim 18. If Claim 18 is rejected, then per MPEP 2144.03, please cite a reference for finding email address to which email is to be sent in case destination is full.

Applicant further notes that each of Claims 15-18 is dependent on Claim 14 and is therefore believed to be patentable for at least the same reasons as Claim 14. Hence Applicant respectfully requests the Examiner to withdraw the prior art rejections of each of Claims 14-18.

Claim 12 was also rejected over the above-quoted text of Bailey's patent. In the second paragraph on page 5 of the Office Action, the Examiner cited to Bailey's column 7 lines 31-67 and column 8 lines 1-65. Furthermore the Examiner stated that Bailey's system checks any available files, and then only available files are transferred for copying. Applicant respectfully traverses. As seen from the text quoted at page 7 of this amendment, Bailey's patent fails to disclose or suggest checking for availability of files, contrary to the Examiner's statement. In fact Bailey's patent doesn't recognize the

problem posed by circular links, and hence doesn't say what, if any, action is to be taken to handle circular links.

Even assuming the Examiner is correct (which isn't true for the above-discussed reasons), there is no indication that the Examiner-suggested checking for a file is to be done by following a link. Following a link during Examiner-suggested checking of a file will result in a problem, as follows. During such checking, the file will appear to be present because the link is present. When the contents of the file to be retrieved for performing the copy, an endless loop will be entered, if a link points to itself (also called "circular" links). Therefore, **the Examiner-proposed modification results in an endless loop.** Claim 12 is therefore believed to be in form for allowance and allowance thereof is respectfully requested. If Claim 12 is rejected, then per MPEP 2144.03, please cite a reference for checking for a circular link before copying.

Claim 13 was also rejected over the above-quoted text of Bailey's patent. In the third paragraph on page 5 of the Office Action, the Examiner cited to Bailey's column 7 lines 31-67 and column 8 lines 1-65. As seen from the text quoted at page 7 of this amendment, Bailey's patent fails to disclose or suggest **string comparison**, contrary to the Examiner's statement. Hence, the Examiner has failed to make a prima facie case of rejection of Claim 13. If Claim 13 is rejected, then per MPEP 2144.03, please cite a reference for performing string comparison to implement checking for a circular link before copying.

Claim 5 was rejected over the teachings of Bailey's patent with the Examiner explaining the rejection in the last paragraph on page 4 of the Office Action. The Examiner cited to Bailey's column 1 lines 61-67, column 2 lines 1-6, column 4 lines 16-45 and column 11 lines 3-6. The cited text is reproduced below:

Column 1 lines 61-67:

In general, in all these cases, there is a problem of maintaining synchronisation between two sets of files: a set of source files and a set of target files. The trivial solution is to delete all the files from the target and copy all the files from the source set. This is reliable but can take a great deal of time if the source set is very large.

Column 2, lines 1-6:

An alternative is to try to identify only the changes that have taken place on the source set and to make the same changes to the target set. It is a relatively simple matter to identify the new files and files whose contents have been changed by examining a modification date (or timestamp) associated with the file and to copy these new or changed files to the target file set.

Column 4 lines 16-45:

The file entries contain a name field and a pointer field PTR which points to the location in storage area 11 of the actual data file, F1 in the illustrated example.

The directory entry for a file--whether normal or anti--is, minimally:

file name
file type Normal .vertline. Anti (N/A)
file size (normal file only)
data ptr (normal files only)
anticount (antifile only)

Returning to FIG. 2, various operations can be performed on file data entries in a directory. As far as relevant to this invention, these are: CREATE, DELETE, COPY, MOVE, RENAME AND LIST. These operations will now be considered for normal files and anti-files.

CREATE is often implicit, in that it involves creating an entry for the file name in the file system in order to read, write or perform other operations on the file. In the case of an anti-file the file system entry for the file has an attribute which indicates its special status. The file becomes an anti-file due to it being deleted as a normal file, the DELETE operation is used with an option that indicates that an anti-file is required. The nature of the anti-file can be transient (it is deleted when it deletes its counterpart normal file) or persistent (it is retained after such deletion takes place).

DELETE removes the file system entry for the file. Normal files and anti-files are handled in exactly the same way. The DELETE for a normal file may be used with an option that requests that an anti-file be created. DELETE for an anti-file simply deletes the anti-file in the directory.

Note that the Examiner's citation of column 11 of Bailey's patent (at the bottom of page 4 of the Office Action) is improper because this column does not exist in Bailey's patent.

As seen by reading the above-quoted text from Bailey's patent, there is no support whatsoever for increasing a limit on a resource as recited in Claim 5.

In fact the word "limit" doesn't appear anywhere in the entirety of Bailey's patent. For this reason, the Examiner is requested to provide a reference which discloses or suggests to increase a limit on a resource and using the resource at the increased limit, as recited in Claim 5.

In the rejection of Claim 5, the Examiner did not provide any motivation whatsoever for increasing a resource's limit. Hence, the Examiner is hereby requested to explain why a skilled programmer would be motivated to increase the limit instead of simply using the resource at its current limit. In view of the above remarks, Applicant submits that Claim 5 is in form for allowance, and allowance thereof is respectfully requested. If Claim 5 is rejected, then per MPEP 2144.03, please cite a reference for changing a resource limit before copying.

Claim 6 was rejected over the teachings of Bailey's patent with the Examiner explaining the rejection in the first paragraph on page 5 of the Office Action. The Examiner cited to Bailey's text which has been reproduced at page 7 of this amendment. As seen from this text, Bailey's patent fails to disclose or suggest that the resource which is being increased (as per Claim 5) is the number of open files. Moreover, there is no indication whatsoever in Bailey's patent that the number of open files will limit his copy operation in any manner, and hence there is no motivation as to why this resource is to be increased. If Claim 6 is rejected, then per MPEP 2144.03, please cite a reference for changing a limit on the number of files before copying.

Claim 7 was rejected over the teachings of Bailey's patent with the Examiner explaining the rejection in the first paragraph on page 5 of the Office Action. The Examiner cited to Bailey's text which has been reproduced at page 7 of this amendment. As seen from this text, Bailey's patent fails to disclose or suggest that the resource which is being increased (as per Claim 5) is the file size. Moreover, there is no indication whatsoever in Bailey's patent that file size will limit his copy operation in any manner, and hence there appears to be no motivation as to why this resource should be increased. If Claim 7 is rejected, then per MPEP 2144.03, please cite a reference for changing a limit on file size before copying.

Claim 1 was rejected over the teachings of Bailey's patent with the Examiner explaining the rejection on page 3 and top half of page 4 of the Office Action. In explaining

the rejection the Examiner stated in the middle of page 3, "Bailey discloses spawning a new process, see (col. 5, lines 58-67 to col. 6, lines 1-67 to col. 7, lines 1-31, col. 2, lines 41-67 to col. 3 lines 1-50)." As seen by reading the cited text from Bailey's patent, there is **no support whatsoever for spawning a new process** as recited in Claim 1. Spawning a new process is a simple concept, namely a new process is started. As seen from the cited text, the Examiner has made a very extended citation. But a careful review of the cited text finds that there is nothing in this text about spawning a new process. Therefore the Examiner's statement is not supported by his citation. If the Examiner continues this rejection the Examiner is respectfully requested to provide a pin-point cite to one sentence or a handful of sentences in Bailey's patent where new process spawning is described.

Moreover, in the rejection of Claim 1, the Examiner stated in the first sentence of the second full paragraph on page 3 of the Office Action that "Bailey discloses, if the item to be copied is a directory, spawning a new process ..." To support his statement, the Examiner cited to the above-quoted text from Bailey's patent. As seen by reading the Examiner-cited text, Bailey is silent on what is to be done if the item to be copied is a directory. In the Examiner-cited text, Bailey merely discusses operations on files, and adding or deleting a file's directory entry. There is no suggestion by Bailey that a new directory store 12 (see top of Bailey's FIG. 1) is to be created, as would happen if a directory were to be copied by Bailey's method. Hence even if Bailey discloses copying a file, **Bailey doesn't disclose copying a directory**. To the extent the Examiner relies on Bailey's FIG. 1's operations "ADD DIR" and "DEL DIR", Applicant submits that these are short-hand notations for operations which are explicitly stated in the cited text as being performed on a file's directory entry (see column 5 lines 62-63) and not on the directory.

Also, in the rejection of Claim 1, the Examiner stated that Bailey's drivers 17 and 18 do file copying and directory copying at the same time. See the middle of page 3 of the Office Action. To support his statement, the Examiner cited to column 5, lines 58-67, column 6 lines 1-67 and column 7 lines 1-31 which have been reproduced above. As seen by reading the Examiner-cited text, Bailey teaches that a file's directory entry is to be checked for existence (as per column 5 line 60) and only after such action by driver 17, the action by driver 18 is taken on a file ("delete action may be taken on a real file..." as per

column 5 line 63). Hence, **Bailey suggests that driver 17 runs before driver 18**, which is contrary to the Examiner's statement that they run simultaneously.

Even assuming that the Examiner is correct in stating that Bailey's drivers 17 and 18 do file copying and directory copying at the same time, the **Examiner's statements support Applicant's position** for the following reason. Specifically, if driver 17 is spawned for a directory, and if driver 18 is spawned for a file, then in Bailey's copy operation these two items (directory and file) are being handled identically (both spawn). Hence there is no suggestion by Bailey to treat a directory differently from file. Specifically there is no suggestion that driver 17 is to be spawned into a new process but driver 18 is to be executed by the current process to simply do copying (i.e. without spawning).

Finally, the Examiner cited to multitasking operating systems designed in the early 1960s (see the bottom of page 3 and top of page 4 of the Office Action). Applicant respectfully submits that **Applicant is not trying to patent just the spawning of a process**. Instead, Applicant's invention is related to conditional spawning in the context of copying of items such as files or directories. Applicant's Claim 1 requires spawning of a new process if the item to be copied is a directory but simply copying the item if the item is a file. For support, see page 5 line 20 of the specification. Applicant submits that 1960s multitasking doesn't disclose or suggest such specific limitations. If the Examiner continues this rejection, Applicant requests the Examiner to supply prior art references to teach Claim 1's conditional spawning if directory but copying if file, per MPEP 2144.03.

The conditional spawning of Claim 1 speeds up copying because, depending on the number of directories to be copied, a corresponding number of processes are created.

Applicant also notes that although two Office Actions issued after filing of the Information Disclosure Statement (IDS) of October 6, 2003, the Examiner failed to consider the cited information. Such failure to consider a properly filed IDS is a USPTO error. Applicant notes that these same references (hereinafter "PCT references") were cited by this same Examiner in a corresponding PCT application based on the current application. Not only has the Examiner failed to cite the PCT references of his own accord in the current prosecution history, but the Examiner has even failed to take into account an IDS filed by Applicant. This **error by the Examiner is glaring** in view of an explicit request made to the Examiner on page 13 of the prior amendment dated October 6, 2003.

If, for some reason, the IDS filed October 6, 2003 is not in the USPTO file, then please call the undersigned before generation of the next Office Action so that alternative arrangements can be made to supply an extra copy of the previously-filed IDS.

Applicant is being forced to file this RCE to address USPTO errors noted above. Hence, Applicant requests that **a thorough and skilled examination** be conducted in response to this RCE. In particular, Applicant respectfully requests that the next Office Action clearly document the USPTO response to each individual argument made in this Amendment as required by MPEP § 707.07(f). In particular, for each claim being rejected, Applicant respectfully requests the USPTO to thoroughly document each limitation that is not explicit in the cited reference (MPEP 2144.03(C)), and also document a prior art motivation for any modification.

In view of the above remarks, Applicant submits that all pending claims are in form for allowance and allowance thereof is respectfully requested. Should there be any questions concerning this paper, please call the undersigned at (408) 982-8200, ext. 3.

Via Express Mail Label No.
EV 448 866 957 US

Respectfully submitted,



Omkar K. Suryadevara
Attorney for Applicant
Reg. No. 36,320